

REMARKS

Favorable reconsideration of this application, as presently amended and in light of the following discussion, is respectfully requested.

Claims 1-9, 11-16, 19, 21-23, and 26 are currently pending. Claims 1 and 22 have been amended by the present amendment. The changes to the claims are supported by the originally filed specification and do not add new matter.

In the outstanding Office Action, Claims 1-5, 22, 23, and 26 were rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent Application Publication No. 2002/007369 to Saravanan (hereinafter “the ‘369 application”) in view of U.S. Patent Application Publication No. 2003/0163372 to Kolsy (hereinafter “the ‘372 application”), further in view of Stonehenge (“Web Techniques Column 56”); Claim 6 was rejected under 35 U.S.C. § 103(a) as being unpatentable over the ‘369 and ‘372 applications, further in view of Stonehenge and U.S. Patent No. 6,865,593 to Reshef et al. (hereinafter “the ‘593 patent”); Claim 7 was rejected under 35 U.S.C. § 103(a) as being unpatentable over the ‘369 and ‘372 applications, further in view of Stonehenge, the ‘593 patent, and U.S. Patent No. 5,875,443 to Nielson (hereinafter “the ‘443 patent”); Claim 8 was rejected under 35 U.S.C. § 103(a) as being unpatentable over the ‘369 and ‘372 patents, further in view of the Stonehenge reference, the ‘593 patent, the ‘443 patent, and U.S. Patent Application Publication No. 2003/0131316 to Brown et al. (hereinafter “the ‘316 application”); Claim 9 was rejected under 35 U.S.C. § 103(a) as being unpatentable over the ‘369 and ‘372 application, further in view of Stonehenge and U.S. Patent No. 6,470,338 to Rizzo et al. (hereinafter “the ‘338 patent”); Claims 11-16 were rejected under 35 U.S.C. § 103(a) as being unpatentable over the ‘369 and ‘372 applications, further in view of Stonehenge and U.S. Patent Application Publication No. 2002/0107699 to Rivera et al. (hereinafter “the ‘699 application”), further in view of U.S. Patent Application Publication No. 2003/0014479 to Shafron et al. (hereinafter

“the ‘479 application”); Claim 19 was rejected under 35 U.S.C. § 103(a) as being unpatentable over the ‘369 and ‘372 applications, further in view of the Stonehenge reference and U.S. Patent Application Publication No. 2002/0037261 to Meffert et al. (hereinafter “the ‘261 application”); and Claim 21 was rejected under 35 U.S.C. § 103(a) as being unpatentable over the ‘369 and ‘372 applications, further in view of the Stonehenge reference and U.S. Patent Application Publication No. 2001/0029521 to Matsuda et al. (hereinafter “the ‘521 application”).

Applicants wish to thank the Examiner for the interview granted Applicant's representative on January 30, 2008, at which time a proposed amendment to the claims was discussed. In particular, the teachings of the Stonehenge reference were discussed. At the conclusion of the interview, the Examiner indicated that further consideration of the prior art would be required in light of the discussion.

Amended Claim 1 is directed to an information providing apparatus for providing prescribed information to a user terminal, comprising: a frame page creator configured to create a frame page requested from the user terminal, the frame page having a plurality of frames, the frame page creator including (1) a loading module configured to extract an argument from a first URL transmitted from the user terminal, create a loading page that contains the extracted argument, the argument being used to specify information to be displayed in one of the frames and a display mode of **another** of the frames, and transmit the created loading page to the user terminal; and (2) a frame page module configured to receive a second URL supplied from the user terminal, the second URL including a second argument based on the argument in the loading page, and to create the frame page having the plurality of frames to be displayed according to the argument in the loading page. The changes to Claim 1 are supported by the originally filed specification and do not add new matter.¹

¹ See, e.g., Figures 13 and 14 and the discussion related thereto in the specification.

Regarding the rejection of Claim 1 under 35 U.S.C. § 103, the Office Action asserts that the '369 application discloses everything in Claim 1 with the exception of the "argument and to create a loading page that contains the extracted argument, the argument being used to specify information to be displayed in one of the frames and a display mode of another of the frames, a second URL supplied from the user terminal based on the loading page, the frame page to be displayed according to the argument and the loading page,"² and relies on the '372 application and the Stonehenge reference to remedy those deficiencies.

The '369 application is directed to a network navigation method including the steps of receiving page instructions for displaying a first web page, wherein the page instructions include frame instructions for displaying an application on the first web page. Further, the '369 application discloses that the network navigation method includes the step of receiving new page instructions for displaying a second web page, determining that the second web page includes the application, and formatting the second web page for display based on the new page instructions and the application instructions, if the second web page includes the application. Applicants note that the '369 application discloses a conventional JAVA frame page.

However, as admitted in the outstanding Office Action, the '369 application fails to disclose a loading page module configured to extract an argument from a first URL transmitted from the user terminal, create a loading page that contains the extracted argument, the argument being used to specify information to be displayed in one of the frames and a display mode of another of the frames, and a frame page module configured to create the frame page to be displayed according to the argument in the loading page, as recited in Claim 1.

² See pages 4 and 5 of the outstanding Office Action.

Further, Applicants respectfully submit that the '369 application fails to disclose a frame page module that is configured to receive a second URL supplied from the user terminal, the second URL including a second argument based on the argument in the loading page, as recited in amended Claim 1.

The '372 application discloses that content frames and advertisements frames can be sent to a user, wherein the advertisement frames can be randomly changed and sent at different times. As shown in Figure 5, the '372 application discloses that the user 505 clicks on a hyperlink in the content frame, which causes a Java script in the content frame to notify the advertisement frame to stop displaying the advertisement in the advertisement frame. Further, the '372 application discloses that the content program module 510 can send a start advertisement display 512 to the advertisement program module 515.

However, as admitted in the outstanding Office Action the '372 patent fails to disclose creating a load page that contains the extracted argument, the argument being used to specify information to be displayed in one of the frames, and a display mode of another of the frames, as recited in amended Claim 1.

Moreover, Applicants respectfully submit that the '372 application fails to disclose a frame page module configured to receive a second URL supplied from the user terminal, the second URL including a second argument based on the argument in the loading page, as recited in amended Claim 1. In this regard, the Office Action refers to paragraph 71 in the '372 application as disclosing this limitation. However, an examination of this section in the '372 application reveals that it relates to the interaction diagram shown in Figure 5. Figure 5 illustrates that the user clicks on a hyperlink, in order to start the display of an advertisement. However, the '372 application does not disclose that a second URL includes a second argument based on the argument in the loading page, as recited in Claim 1. Assuming that the Office Action is equating the claimed argument with the requested frame in the '372

system, Applicants note that the '372 system merely discloses the request for a target web page and a request for an advertisement, but does not disclose that a second argument is based on the argument in the loading page, as recited in Claim 1.

The Stonehenge reference is directed to a chat server and a chat client written in the Perl language. In particular, the Office Action has cited to lines 53-74 in a program listing of the web chat server. The Stonehenge reference discloses that lines 53-65 of the program handle the "read" window, while lines 66-80 handle the "write" window. In this regard, Applicants note that the Office Action states that the argument recited in Claim 1 is given by the "code" variable disclosed in lines 48, 53, and 66 of the listed program. However, line 48 of the program indicates that the path being requested from the web server is extracted from a URL and assigned to the code variable. Page 5 of the Stonehenge reference indicates that the code variable is then used as a branching point to determine whether there has been a request to read the chat log or a request to write a message to the chat log.

However, Applicants respectfully submit that the Stonehenge reference fails to disclose an argument being used to specify information to be displayed in one of the frames, and a display mode of **another** of the frames, as required by Claim 1. In this regard, Applicants note that the Stonehenge reference discloses a read frame and a write frame, wherein the read frame is a frame in which past messages are posted, and the write frame is a frame in which a user can enter a new message. Further, page 4 of the Stonehenge reference indicates that the "read10 URL" creates a default refresh rate of once every ten seconds, which can be changed later by the user. The Stonehenge reference does not indicate that there is any refresh rate parameter that can be associated with a write window. Applicants note that lines 56-58 are directed to allowing the user to change the refresh rate of the read window.

However, Applicants note that the Stonehenge code parameter merely controls whether processing is done to the read frame or to the write frame, and in the case of a read, might also control the initial refresh rate of the read frame, but is not an argument that specifies information to be displayed in one of the frames and a display mode of **another** of the frames. If the code parameter disclosed by the Stonehenge et al. reference is a read parameter, it does not specify the display mode of the write frame, as is required by Claim 1. Moreover, if the code parameter is a write parameter, it is not an argument that is used to specify a display mode of the read frame, as would be required by Claim 1. Thus, contrary to the assertion on page 21 of the outstanding Office Action, the code variable disclosed by the Stonehenge et al. reference is not equivalent to the argument that is used to specify information to be displayed in one of the frames, and a display mode of **another** of the frames, as required by Claim 1.

Thus, no matter how the teaching of the '369 application, the '372 application, and the Stonehenge reference are combined, the combination does not teach or suggest a loading page module configured to extract an argument from a first URL transmitted from the user terminal, create a loading page that contains the extracted argument, the argument being used to specify information to be displayed in one of the frames and a display mode of another of the frames, as well as the second URL including a second argument based on the argument in the loading page, recited in Claim 1. Accordingly, Applicants respectfully submit that amended Claim 1 (and all similarly rejected dependent claims) patentably define over any proper combination of the '369 application, the '372 application, and the Stonehenge reference.

Independent Claim 22 recites loading page means for creating a loading page that contains an extracted argument, the argument being used to specify information to be displayed in one of the frames, and a display mode of another of the frames. As discussed

above, the combined teachings of the '369 application, the '372 application, and the Stonehenge reference fail to disclose this limitation. Independent Claim 23 recites similar limitations. Accordingly, Applicants respectfully submit that independent Claims 22 and 23 patentably define over any proper combination of the '369 application, the '372 application, and the Stonehenge et al. reference.

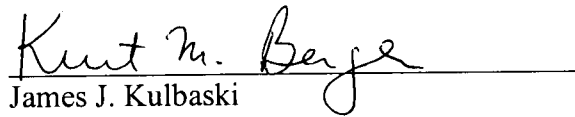
Regarding the rejection of dependent Claims 6-9, 11-16, 19, and 21 under 35 U.S.C. § 103, Applicants respectfully submit that the '593 patent, the '443 patent, the '316 application, the '338 patent, the '699 application, the '479 application, the '261 application, and the '521 application fail to remedy the deficiencies of the '369 application, the '372 application, and the Stonehenge et al. reference, as discussed above. Accordingly, Applicants respectfully submit that dependent Claims 6-9, 11-16, 19, and 21 patentably define over the cited references.

Thus, it is respectfully submitted that independent Claims 1, 22, and 23 (and all associated dependent claims) patentably define over any proper combination of the cited references.

Consequently, in view of the present amendment and in light of the above discussion, the outstanding grounds for rejection are believed to have been overcome. The application as amended herewith is believed to be in condition for formal allowance. An early and favorable action to that effect is respectfully requested.

Respectfully submitted,

OBLON, SPIVAK, McCLELLAND,
MAIER & NEUSTADT, P.C.

A handwritten signature in cursive script, appearing to read "Kurt M. Berger", is written over a horizontal line.

James J. Kulbaski
Attorney of Record
Registration No. 34,648

Customer Number

22850

Tel: (703) 413-3000
Fax: (703) 413 -2220
(OSMMN 03/06)

Kurt M. Berger, Ph.D.
Registration No. 51,461